

# 5AMU3L 5AY5

Brian Mount

```
000002000112000012111102000002111002001112111112000002000
112000012000002000002100002000012000002000012020011121111
120000120000020000121111020000121100020000121110020000020
000120011121111120000120200001211110200001202000012000002
001112111112000002000012000002110002000012001020011121111
120000120200001211110200001202000012000002001112111112
```

As the title hints, this is a variation on Morse code. Use: 0 => dot, 1 => dash, and 2 => pause.

Converting to Morse Code:

```
..... -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-.
-.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-.
-.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-.
-.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-.
-.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-.
-.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-. -.-.-.
```

Convert using Morse code:

```
53495820534556454E204549474854204E494E452054574F204E494E4520
```

Convert using ASCII (interpreting each 2 byte as hex):

```
SIX SEVEN EIGHT NINE TWO NINE
```

Converting to numerals:

```
678929
```

Converting back to Hexadecimal (given the 0x):

```
A5C11
```

Answer: **ASCII**